06-16-00

A

Certificate of Mailing By "U.S. Express Mail" Under 37 C.F.R. 1.10(c)
"EXPRESS MAIL" Mailing Label Number. EL523936592US Date of Deposit: 6/14/2000
I hereby certify that this paper and/or fee is being deposited with the United States Postal Service "EXPRESS MAIL POST OFFICE TO ADDRESSEE" service under 37 C.F.R. 1 10 on the date indicated above and is addressed to the Assistant Commissioner For Patents, Washington, DC 20231.
Name: Mary-Charmaine Gorgonio
Signature: Mary C. Gorg
Signature Date: 6/14/2000

Docket No. : SIA-P008

APPLICATION TRANSMITTAL LETTER

Assistant Commissioner of Patents
United States Patent and Trademark Office
Washington, D.C. 20231
ATTN: BOX PATENT APPLICATION

Sir:

Transmitted herewith for filing is the patent application of
Inventor(s): **HUIE, J., *et al.***
Entitled: **A HIGH PERFORMANCE NETWORK ADDRESS PROCESSOR SYSTEM**
16 No. pages of specification, including title page, claims and abstract
3 No. sheets of X informal, ____ formal drawings

Also enclosed are:
X Executed Combined Declaration and Power of Attorney for Patent Application
X An Original Executed Assignment of the Application
X Form PTO-1595 (Recordation Cover Sheet for Assignment)
X Verified Statement Claiming Small Entity Status with Cover Sheet
___ An Information Disclosure Statement (Form PTO-1449A and Form PTO-1449B)
___ A copy of References cited in Information Disclosure: ________ documents

FEES DUE
The fees due for filing the application pursuant to 37 C.F.R. 1.16 and for recording the Assignment, if any, are determined as follow:

CLAIMS					
	No. of Claims		Extra Claims	Rate	Fees
Basic Application Fee ($690 large entity; $345 small entity)					$ 345.00
Total Claims	9	Minus 20 =	0	X $18 = X $ 9 (small) =	.00
Total Independent Claims	3	Minus 3 =	0	X $78 = X $39 (small) =	.00
If Multiple Dependent Claims are presented, add $260.00 or $130.00(small)					
If Assignment enclosed, add Assignment Recording Fee $40.00					40.00
TOTAL APPLICATION FEE DUE					$ 385.00

PAYMENT OF FEES
The full fee due in connection with this communication is $ 385.00
and is provided as follows:

_____ The Commissioner is hereby authorized to charge the fees associated with this communication or credit any overpayment to **Deposit Account No: 500482**. A duplicate copy of this authorization is enclosed.

_X_ A Check No. 3555 for the above specified full fee is enclosed. However, in case Applicant inadvertently miscalculated any required fee, the Commissioner is hereby authorized to charge the necessary additional amount associated with this communication or credit any overpayment to **Deposit Account No: 500482**. A duplicate copy of this authorization is enclosed.

This application is filed pursuant to 37 C.F.R. 1.53 in the name of the above-identified Inventor(s).

Please direct all correspondence concerning the above-identified application to the following address:

FERNANDEZ & ASSOCIATES LLP
PATENT ATTORNEYS
PO BOX D
MENLO PARK, CA 94026-6204

(650) 325-4999
(650) 325-1203 : FAX
EMAIL: *iploft@iploft.com*

022877

PATENT TRADEMARK OFFICE

Respectfully submitted,

IRENE H. FERNANDEZ, ESQ.
Reg. No. 34,625

6-14-2000
Date

The Assistant Commissioner of Patents
United States Patent and Trademark Office
Washington, D.C. 20231
ATTN: Box Patent Application

Re: U.S. Utility Patent Application
Appl. No. (Not yet assigned); Filed 6/14/2000
For: **A HIGH PERFORMANCE NETWORK ADDRESS PROCESSOR SYSTEM**
Inventor(s): **HUIE, J., *et al.***
Docket No.: **SIA -P008**

Sir:
The following documents are forwarded herewith for action by the U.S. Patent and Trademark Office:

1. U.S. UTILITY APPLICATION
entitled: **A HIGH PERFORMANCE NETWORK ADDRESS PROCESSOR SYSTEM**
having named inventor(s):
HUIE, J., *et al.*
   a. a specification consisting:
      (i) 1 pages prior to the claims, including title page;
      (ii) 2 pages of claims;
      (iii) 1 page abstract;
   b. 3 sheets of **informal** drawings: (FIGs. 1, 2, 3);
2. An **original, executed** Combined Declaration and Power of Attorney by named inventors;
3. Form PTO-1082 (in duplicate);
4. Cover letter for Assignment (Form PTO-1595)
5. An original, executed Assignment to **SILICON ACCESS NETWORKS**, executed by named inventors, recordation of which is hereby requested;
6. A return post card; and
7. Check No. 3555 for $ 385.00 to cover:

Patent application filing fee:	$ 345.00
Assignment Recordation fee:	$ 40.00
Excess claims fee:	$ .00

8. Verified Small Entity Status Statement with Cover Sheet

It is respectfully requested that the attached postcard be stamped with the filing date of the above documents and unofficial application number and returned to the addressee as soon as possible.

Respectfully submitted,

6-14-2000
Date

IRENE H. FERNANDEZ, ESQ.
Reg. No. 34,625

FERNANDEZ & ASSOCIATES LLP
PATENT ATTORNEYS
PO BOX D
MENLO PARK, CA 94026-6204
(650) 325-4999
(650) 325-1203 : FAX
EMAIL: *iploft@iploft.com*

Docket No.: **SIA-P008**

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application No.: (Not yet assigned) Patent No.: (Not yet assigned)
Filed on: 6/14/2000 Issued on: (Not yet assigned)

Title: **A HIGH PERFORMANCE NETWORK ADDRESS PROCESSOR SYSTEM**
Inventor(s): **HUIE, J.,** ***et al.***

VERIFIED STATEMENT CLAIMING SMALL ENTITY STATUS

(37 CFR 1.9(f) and 1.27(c) -- SMALL BUSINESS CONCERN)

I hereby declare that I am

____ the owner of the small business concern identified below:

_X_ an official of the small business concern empowered to act on behalf of the concern identified below:

Name of Small Concern:
SILICON ACCESS NETWORKS

Address of Small Concern:
2801A ORCHARD PARKWAY
SAN JOSE, CA 95134-2013

I hereby declare that the above identified small business concern qualifies as a small business concern, as provided in 37 CFR 1.9(d), for purpose of paying reduced fees to the United States Patent and Trademark Office, in that the number of employees of the concern, including those of its affiliates, does not exceed 500 persons.

I hereby declare that the rights under contract or law have been conveyed to, and remain with, the concern identified above with regard to the invention described in

_X_ the specification filed herewith, with title and inventor(s) as listed above.
____ the application identified above.
____ the patent identified above.

If the rights held by the above identified concern are not exclusive, each individual, concern, or organization having rights in the invention is listed below and no rights to the invention are held by any person, other than the inventor, who would not qualify as an independent inventor under 37 CFR 1.9(c), it that person made the invention, or by any concern which would not qualify as a small business concern under 37 CFR 1.9(d), or a nonprofit organization under 37 CFR 1.9(e).

Each person, concern or organization having any rights in the invention in addition to the above identified concern is listed below:
Name ____________________
Address ____________________
____ Individual _X_ Small Business ____ Nonprofit Organization

I acknowledge the duty to file, in this application or patent, notification of any change in status resulting in loss of entitlement to small entity status prior to paying, or at the time of paying, the earliest of the issue fee

or any maintenance fee due after the date on which status as a small business entity is no longer appropriate. (37 CFR 1.28(b))

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true and further, that these statements were made with knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application, any patent issuing thereon, or any patent to which this verified statement is directed.

PERRY CONSTANTINE
Print Name of Person Signing

PRESIDENT AND CHIEF EXECUTIVE OFFICER
Title of Person Signing

SILICON ACCESS NETWORKS
2801A Orchard Parkway, San Jose, CA 95134-2013
Address of Person Signing

SIGNATURE ____________ Date 6-12-00

Application

For

United States Utility Patent

Title:

A High Performance Network Address Processor System

Inventor(s):

Jonathan Huie, residing at 65 E. Latimer Ave, Campbell, CA 95008, a citizen of the United States, and

James Michael O'Connor, residing at 1877 Slate Drive, Union City, CA, 94587 (residence), a citizen of the United States.

A High Performance Network Address Processor System

Field of Invention

This invention relates to internet protocol routing processors, particularly to efficient internet protocol search engines.

Background of Invention

The most common bottleneck to fast internet protocol (IP) routing processors resides in having a fast and efficient IP searching method and to speed up forwarding operations. Typical IP search engines and network address processors provide a single central routing table such as provided by the Internet Protocol Routing Processor, IPRP-V4 and its associated family of Internet Protocol Routing Processor, provided by Alliance Semiconductor of Santa Clara, California. The IPRP family of network address processors operate at a typical frequency of 66MHz and upon receipt of a given 32-bit wide destination address, the network address processor searches its central lookup table for a matching entry, i.e., an entry matching the largest number of higher order bits of the destination address according to the CIDR protocol. When greater than 16-bits addressing is required for routing an IP packet, a 16-bit output from the network address processor is used as a pointer to an external memory for further address matching searching. Accessing an external memory consequently adds to processing time and cost.

Thus, with the ever increasing and escalating consumer demand and usage of internet applications there is a need for a network address processor to handle faster and greater network address routing requirements to avoid internet traffic bottlenecks.

Summary of Invention:

A high performance network address processor (NAP) is provided comprising a longest prefix match lookup engine for receiving a data lookup request and in response thereto provide a key and data pointer address to an associated data engine. The associated data

engine in response thereto then provides a NAP data output associated with a designated network destination address requested. The high performance NAP longest prefix match lookup engine comprises a plurality of pipelined lookup tables, each table of a predetermined size, calculated based on the number of entries in the next higher sequenced table. Each lower level table thus provides an index to a given row within the next higher stage lookup table. The output of the longest prefix match lookup engine comprises an associated data pointer provided as input to the associated data engine. The associated data engine also comprises one or more lookup tables, wherein the associated data engine generates a designated data output in response to the output of a longest prefix match lookup engine. The associated data engine also comprises a plurality of update functions associated with each lookup table. These update functions modify the designated data output based on a plurality of data fields provided to the high performance network address processor with the data lookup request.

The high performance network address processor thus allows on-the-fly modification of indexed user data, and also offloads CPU intensive operations from other processors in the system. For applications with longest-prefix match searches (including CIDRs), a NAP-based solution achieves greater densities than other TCAM-based approaches. Moreover, the NAP provides lower power dissipation.

Brief Description of Drawings

Fig. 1 is a general block diagram illustrating a network address processor in a typical system application according to the principles of this invention.

Fig. 2 is a more detailed block diagram of the network address processor shown in Fig. 1.

Fig. 3 is a more detailed block diagram of the longest prefix match lookup engine and the associated data engine of the network address processor illustrated in Figs. 1 and 2.

Detailed Description of Preferred Embodiment(s)

Fig. 1 illustrates a general block diagram illustrating a network address processor (NAP) 10 according to the principles of this invention. High performance NAP 10 comprises a longest prefix match lookup engine 20 for receiving a request for data 40 of a designated network destination address. An associated data engine 30 is also provided and coupled to the longest prefix match lookup engine 20 for receiving a longest prefix match lookup engine output 22 comprising an address pointer and key from the longest prefix match lookup engine 20. The associated data engine provides in response thereto a network address processor data output 43 corresponding to the designated network destination address provided to NAP 10.

The high performance network address processor longest prefix match lookup engine 20 comprises a plurality of pipelined lookup tables as described in greater detail relative to Figs. 2-3. Each lower stage provides an index to a given key within the next higher stage lookup table. In the preferred embodiment, associated data engine 30 also comprises one or more associated tables and generating as an output, an associated designated data output 43 corresponding to the requested network address 40 provided to NAP 10.

In the preferred embodiment, NAP 10 is designed to allow a large number of pipelined lookup requests to access and update data associated with a given key. The core of the NAP 10 comprises a pipelined longest prefix match lookup engine 20, and an associated data engine 30 comprising one or more associated lookup tables for storing and reading associated data. Keys are maintained with associated masks. These masks indicate the number of least-significant "don't-care" bits within a key and allow multiple range entries with overlapping endpoints to be maintained within the NAP.

Fig. 2 illustrates a more detailed block diagram of the network address processor shown in Fig. 1. In the preferred embodiment, NAP 10 comprises a first interface 52 (interface A) and a second interface 54 (interface B). Each of interfaces A and B comprises an associated ZBT SSRAM interface coupled to receive associated out-of-band signals, i.e., interface A is coupled to routing and management CPU 46 and interface B is coupled to a classification and forwarding ASIC 47. Interfaces A and B also provide packet assembly functions, such as collecting an address lookup request sent using multiple bus cycles. Similarly, each interface A and B also partitions output data requiring several bus requests to complete data transfer.

As an illustration of the operations of NAP 10, a lookup address request 41 received via interface B from classification and forwarding ASIC 47 of Fig 1 is provided to a request decoder 56. Request decoder 56 then determines if the type of request received is a lookup request, a management request, or a high-level command request, and in response thereto to stack the received address with the appropriate FIFO, i.e., provides a lookup address 41 to lookup request FIFO 59. Request decoder 56 also arbitrates between interfaces A and B so that if both interfaces present a request on the same cycle, only one will be serviced. If request handles are not used, then request decoder 56 assigns request numbers as well.

Preferably, a high-level command FIFO, a management request FIFO, and a lookup request FIFO are provided for greater flexibility in network addressing applications. Thus, with three independent FIFOs, each FIFO holds decoded requests to be serviced. Consequently, lookup rate manager can allocate priority to control for every cycle, to choose whether to service a request from the Lookup Request FIFO or the Management Request FIFO. Optimally, rate manager makes this choice based on the values defined by configurable registers so as to guarantee service to a certain rate of lookup requests.

Fig 3. illustrates the pipelined longest prefix match lookup engine 20 in greater detail. Pipelined longest prefix match lookup engine comprising one or more pipelined lookup tables 21, 22, 23, and 24 services all lookups and low-level management requests. First lookup stage 21 (L0), second lookup stage 22 (L1), third lookup stage 23 (L2), and fourth lookup stage 24 (L3) these stages are used to index a given key within the L4 table 25 of an associated data engine 30. L4 Stage 25 comprises a set of keys and their associated data pointers stored within entries of L4. L4 data stage 25 comprises the per-route data, and a pointer to an associated L5 data. Read-modify-write operations take place on data entries selected at this stage. L5 data stage 26 comprises the final stage of the lookup pipeline comprising longest prefix match lookup engine 20 and associated data engine 30. Data entry selected from L4 stage 25 comprises per hop/per RMON/per customer data. L4 and L5 stages also provide the read-modify-write operations.

Result Buffer 62 collects all the associated status and data provided from longest prefix match lookup engine 20 and associated data engine 30 in response to a request 41 that flows through longest prefix match lookup engine 20 or the associated data engine 30. Interfaces A and B responding to read requests receive of their data from here. High Level Engine 60 provides high-level operations, such as insert and delete, using multiple low-level management requests.

The following Table 1 illustrates a sample implementation of stages L0-L5.

Table 1: Storage Levels

Level	**# Entries**	**Content**		
L0	31	144-bit Key	8-bit Mask	
L1	480	144-bit Key	8-bit Mask	
L2	7680	144-bit Key	8-bit Mask	
L3	128K	144-bit Key	8-bit Mask	17-bit Associated Data ID
L3 per bucket	8K	5-bit Count	32 (8-bit Mask, 17-bit Associated Data ID) pairs	

L4 User Data	128K	96-bit Data		
L5 User Data	8K	256-bit Data		

Table 2 below illustrates a sample implementation of memory organization of the various stages of longest prefix match lookup engine 20 and associated data engine 30.

Table 2: Memory Organizations

	Width	Height	Size	Compares
L0	4712 bits	1	4.6 Kb SRAM	31 144-bit
L1	2280 bits	32	71.25 Kb SRAM	15 144-bit
L2	2280 bits	512	~1.11 Mb SRAM	15 144-bit
L3	3509 bits	8 K	~27.4 Mb DRAM	48 144-bit
L4 Data	96 bits	128 K	12 Mb DRAM	
L5 Data	256 bits	8 K	2 Mb DRAM	
Total			**~1.2 Mb SRAM** **~41.4 Mb DRAM**	**109 144-bit**

Lookup Pipeline

Fig. 3 depicts the essential operation of the lookup pipeline 20. Each stage is further described below. The input to lookup pipeline 20 is a key that is to be looked up and provided at either input 40 or 41 (see Fig. 2) to NAP 10. The output of the pipeline is the following associated with the highest priority (longest prefix) match to the supplied search key. A pointer to L4 memory 25 points to a content of the referenced 96-bit L4 memory word, and the contents of the referenced 256-bit L5 memory word. Some embodiments may have some subset of these outputs; some may specify others - such as the matching key, the matching priority, and the address of the matching L3 item. Additional status bits indicating success, failure, or other conditions may also be provided.

L0 Memory

This memory comprises a row of 31 key/mask pairs. Each lookup request reads all 31 entries from the L0 memory. Preferred embodiment is that all 31 pairs are read in a single cycle. Alternatively, all 31 pairs can be read sequentially, but this could limit the rate at which requests could be serviced. Preferably, entries in this memory are sorted, and reflect the maximum key that would be stored in the 16th position in the corresponding row of the L1 Memory.

L0 Comparators/Priority Encoder/Next Address Generator

The 31 keys with associated masks, along with the search key are provided as input to this stage. This stage selects the smallest entry that is greater than or equal to the input search key. If multiple entries have the same key, the key with the smallest mask is selected. The position of the selected element is the output of this stage. Thus, in this embodiment, a 5-bit value indicating which of the 31 elements was selected is passed to the L1 stages. If no entries are greater than the supplied key, the index passed to the L1 memory is the value 32.

L1 Memory

This memory is logically organized as 32 rows of 15 key/mask pairs. Based on the 5 bits generated in the previous stage L0, one row of the L1 Memory is selected. Each lookup request reads all 15 entries from the selected row. Preferred embodiment is that all 15 are read in a single cycle. Alternatively, all 15 entries may be read sequentially, but this could limit the rate at which requests could be serviced. Similar to L0, the entries in this memory preferably are sorted, and reflect the maximum key stored in the corresponding row of the L2 Memory.

L1 Comparators/Priority Encoder/Next Address Generator

The 15 keys, with associated masks, along with the search key and the address generated from the previous stage L0 are the input to this stage. This stage selects the smallest entry

that is greater than or equal to the input search key. If multiple entries have the same key, the key with the smallest priority is selected. Again, if no entries are greater than the supplied key, the 4-bit value of the selected entry is the value 16. The position of the selected element, combined with the address of the row selected by the L0 stage, is the output of this block. Thus, in this embodiment, the 5-bit address from the output of the L0 stage is concatenated with the 4-bit value indicating which of the 16 elements was selected. The resulting 9-bit address is passed to the L2 stage.

L2 Memory

This stage is logically organized as 512 rows of 15 key/mask pairs. Based on the 9 bits generated in the previous stage L1, one row of the L2 Memory is selected. Each lookup request reads all 15 entries from the selected row. Preferred embodiment is that all 15 entries are read in a single cycle. However, similar to L0 and L1, the 15 entries of L2 can be read 15 sequentially, but this could limit the rate at which requests could be serviced. The entries in this memory must be sorted, and reflect the maximum key stored in the corresponding row of the L3 Memory.

L2 Comparators/Priority Encoder/Next Address Generator

The 15 keys with associated priorities, along with the search key and the address generated from the previous stage L2 are the input to this stage. This stage selects the smallest entry that is greater than or equal to the input search key. If multiple entries have the same key, the key with the smallest priority is selected. If no entries are greater than the supplied key, the 4-bit value of the selected entry is the value 16. The position of the selected element, combined with the address of the row selected by the L1 stage, is the output of L2. Thus, in this embodiment, the 9-bit address from the output of the L1 stages is concatenated with the 4-bit value indicating which of the 16 elements was selected. The resulting 13-bit address is passed to the L3 stage.

L3 Memory

This memory is logically organized as 8,192 rows of 32 key/mask/pointer tuples. In addition, each row has 32 mask/pointer pairs and a 6-bit count. Based on the 13 bits generated in the previous block, one row of the L3 Memory is selected. Each lookup request reads all 32 entries from the selected row, plus the count and mask/pointer pairs. The preferred embodiment comprises that all data in the row are read in a single cycle. However, all the data elements may be read sequentially, but this could limit the rate at which requests could be serviced. The entries in each row of this memory are preferably sorted.

L3 Comparators/Priority Encoder/Next Address Generator

The 16 keys with associated masks and L4 data pointers and the other data from a row of the L3 Memory, along with the search key are the input to L3. This stage selects the smallest entry that equals to the input search key with the corresponding number of mask bits ignored. If multiple entries have the same key, the key with the smallest mask is selected. If no keys match the above requirements, the maximum key in the row is compared with the input search key using each of the 32 masks from the 32 mask/pointer pairs. The pointer is selected that corresponds to the smallest mask for which the input search key equals the maximum key in the row with the corresponding number of mask bits ignored. The L4 data pointer associated with the selected element is the output of this block. Thus, in this embodiment, a 17-bit pointer is passed to the L4 stage and provided as an output of the longest prefix match lookup engine 20.

L4 Memory

This memory is logically organized as 131,072 rows of 96 bits. Based on the 17 bits generated in the previous block, one row of the L4 Memory is selected. The 96-bits of the selected row are read from the memory and provided as an output of the lookup and to the L4 update logic. Accordingly, 13 bits of the 96 bits are passed to the L5 memory block.

L4 Update Logic

Associated with each lookup request, various fields of the 96-bit L4 data may be updated and written back to the L4 memory. For instance, the 96-bit data may represent a 20-bit time stamp, a 28-bit packet counter, and a 35-bit byte counter, in addition to the 13-bit pointer to the L5 data. Each of these fields may be updated as a result of a given lookup. This logic could be fixed, or it could be configurable by the user.

L5 Memory

This memory is logically organized as 8,192 rows of 256 bits. Based on the 13 bits read from the L4 memory, one row of the L5 Memory is selected. The 256 bits of the selected row are read from the L5 memory and provided as an output of the lookup, and 128 bits of the 256 bits are provided to the L5 update logic.

L5 Update Logic

Associated with each lookup request, various fields in 128 bits of each 256-bit L5 data element may be updated and written back to the L5 memory. Each of these fields may be updated as a result of a given lookup. This logic could be fixed, or it could be configurable by the user.

Organization of Memory Levels

Preferably, the exact sizes and organizations of the search memories can be configured according to the following rules:

- N = Number of keys - A given number of total keys to be in final level of search memory
- M = Maximum nesting – Specifies the maximum supported depth of ranges which can be nested within other ranges.
- L = Final level of search memory (e.g. 3 for the current example).

- W_L = Level L width - A given maximum number of keys from level L to be examined at once (or in sequence) for a given lookup request. We call this number of elements one row of the level L memory.
- W_{L-1} = Level L-1 width - A given maximum number of keys from level L-1 to be examined at once (or in sequence) for a given lookup request. We call this number of elements one row of the level L-1 memory.
- W_{L-2} = Level L-2 width - A given maximum number of keys from level L-2 to be examined at once (or in sequence) for a given lookup request. We call this number of elements one row of the level L-2 memory.

.

.

.

W_0 = L0 width - A given maximum number of keys from L0 to be examined at once (or in sequence) for a given lookup request. We call this number of elements one row of the level 0 memory.

First, the number of keys and number of levels are chosen. The various width values of each level of memory are determined by various implementation factors. The preferred embodiment is that N and W_L are powers of two (e.g. 2,4,8,16,32,64,...) and W_{L-1}, W_{L-2}, ..., and W_0 are powers of two minus one (e.g. 1,3,7,15,31,63...). Rows are only a logical concept, and not necessarily the physical structure of memory.

The following values can them be computed, **wherein R represents the number of rows**:

- Number of rows of level L – $R_L = N/W_L$
- Number of rows of level L-1 – $R_{L-1} = R_L/(W_{L-1}+1)$

- Number of rows of level L-2 – $R_{L-2} = R_{L-1}/(W_{L-2} + 1)$
- (etc.)

All of these values must be chosen subject to the constraint that in the end, the following values result.

- Number of rows of L0 – $R_0 = R_1/(W_0+1) = 1$

The total size of each memory is:

- $S_L = R_L$ * (((W_L * (key size + mask size + pointer size)) + M * (mask size + pointer size) + $\log_2(W_L)$ + 1), for level L.
- $S_x = R_x * W_x$ * (key size + mask size), for all $x<L$.

Foregoing described embodiments of the invention are provided as illustrations and descriptions. They are not intended to limit the invention to precise form described. In particular, it is contemplated that functional implementation of invention described herein may be implemented equivalently in hardware, software, firmware, and/or other available functional components or building blocks. Other variations and embodiments are possible in light of above teachings, and it is thus intended that the scope of invention not be limited by this Detailed Description, but rather by Claims following.

Claims

What is claimed is:

1. A high performance network address processor comprising:

a longest prefix match lookup engine for receiving a network address request having a designated network destination address; and

an associated data engine coupled to the longest prefix match lookup engine for receiving a longest prefix match lookup engine output address and providing a network address processor data output corresponding to the designated network destination address.

2. The high performance network address processor of claim 1 wherein the longest prefix match lookup engine comprises a plurality of pipelined lookup tables.

3. The high performance network address processor of claim 1 wherein the network address processor is configurable to a variety of destination address width.

4. The high performance network address processor of claim 1 wherein the network address processor generates a network address data output in one clock cycle.

5. A high performance network address processor integrated circuit, wherein the network address processor integrated circuit comprises:

a longest prefix match lookup engine for receiving a network address request having a designated network destination address; and

an associated data engine coupled to the longest prefix match lookup engine for receiving a longest prefix match lookup engine output address and providing a network address processor data output corresponding to the designated network destination address.

6. The high performance network address processor of claim 5 wherein the longest prefix match lookup engine comprises a plurality of pipelined lookup tables.

7. The high performance network address processor of claim 2 wherein the plurality of pipelined lookup tables is implemented in a DRAM.

8. A high performance network addressing method comprising the steps of:

providing a longest prefix match lookup engine with a network address data request and a destination network address, wherein the longest prefix match lookup engine comprises a set of lookup tables;

searching the set of lookup tables to select a look up engine address output from the set of lookup tables to provide to an associated data engine; and

searching the associated data engine to provide an associated destination address data output.

9. The high performance network addressing method of claim 8 wherein the step of searching the set of lookup tables comprises searching for an entry of the set of lookup tables that comprises the smallest entry that is greater than or equal to an input search key, the step of searching for the smallest entry comprising the steps of: selecting the smallest entry that equals the input search key with a corresponding number of mask bits,

wherein if one or more entries comprise the same key, the key having the smallest mask is selected, and

wherein if no key matches the above requirements, the maximum key in a row is compared with the input search key using each of a set of mask pointer pairs, each of the pointer is selected to correspond to the smallest mask for which the input search key equals the maximum key in the row with the corresponding number of mask bits ignored.

Abstract

A high performance network address processor is provided comprising a longest prefix match lookup engine for receiving a request for data from a designated network destination address. An associated data engine is also provided to couple to the longest prefix match lookup engine for receiving a longest prefix match lookup engine output address from the longest prefix match lookup engine and providing a network address processor data output corresponding to the designated network destination address requested. The high performance network address processor longest prefix match lookup engine comprises a plurality of pipelined lookup tables. Each table provides an index to a given row within the next higher stage lookup table, wherein the last stage, or the last table, in the set of tables comprises an associated data pointer provided as input to the associated data engine. The associated data engine also comprising one or more tables, the associated data engine generating a designated data output associated with the designated network address provided to the high performance NAP.

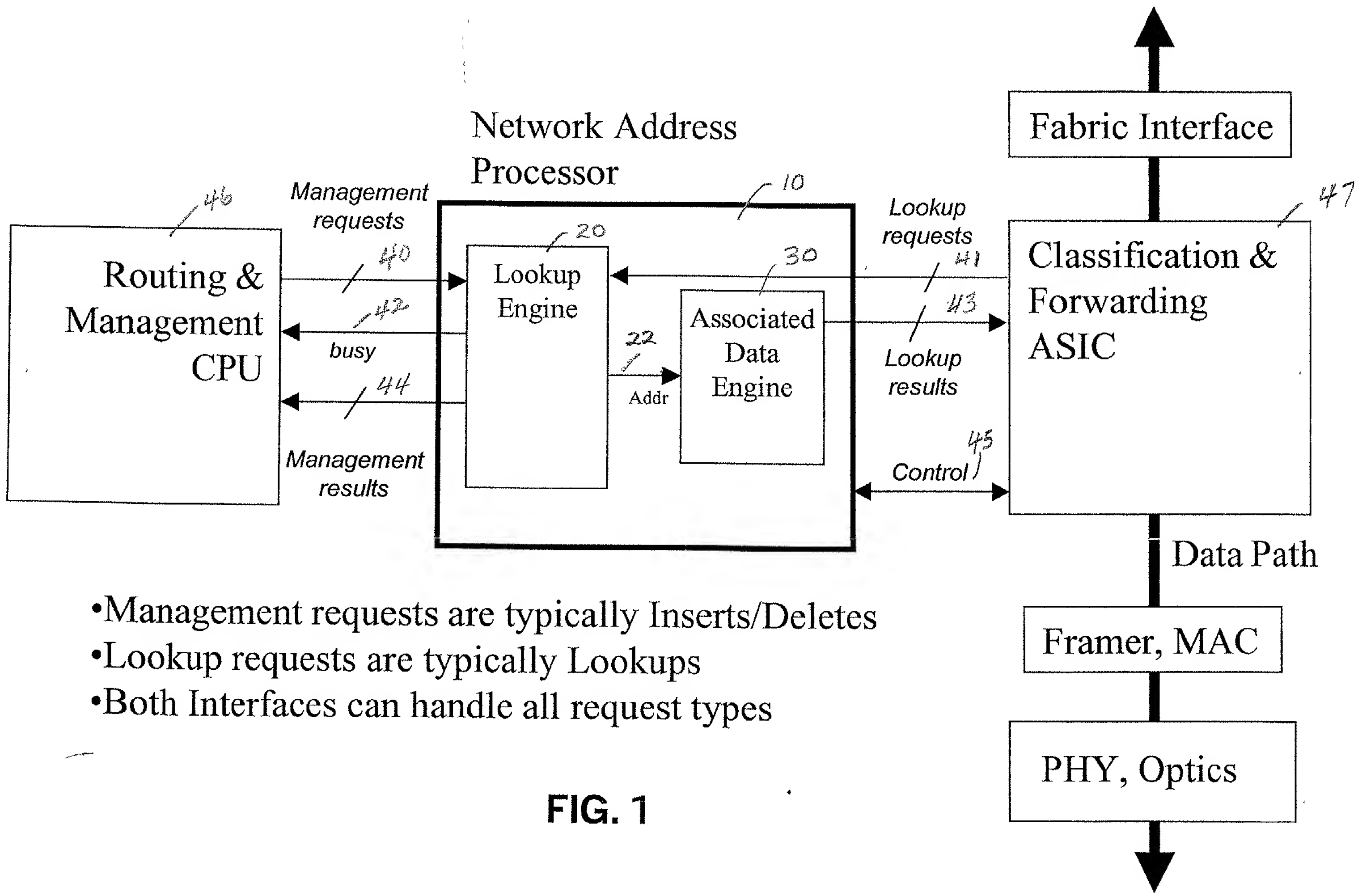
Network Address Processor
10
Routing & Management CPU
46
Management requests
40
42
busy
44
Management results
Lookup Engine
20
22
Addr
Associated Data Engine
30
Lookup requests
41
43
Lookup results
45
Control
Fabric Interface
Classification & Forwarding ASIC
47
Data Path
Framer, MAC
PHY, Optics
•Management requests are typically Inserts/Deletes
•Lookup requests are typically Lookups
•Both Interfaces can handle all request types

FIG. 1

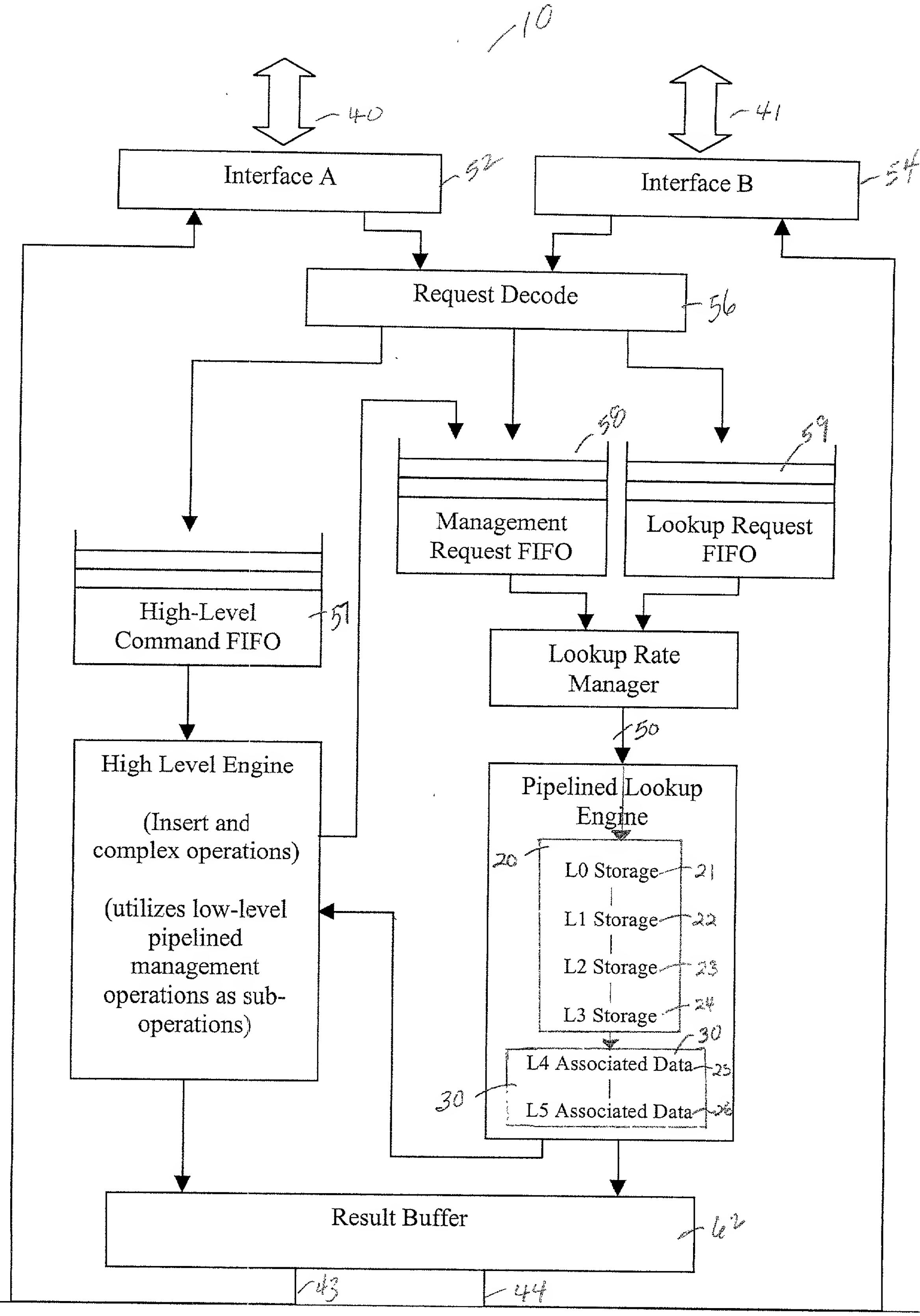
10
40
41
Interface A
52
Interface B
54
Request Decode
56
58
59
Management Request FIFO
Lookup Request FIFO
High-Level Command FIFO
51
Lookup Rate Manager
50
High Level Engine
(Insert and complex operations)
(utilizes low-level pipelined management operations as sub-operations)
Pipelined Lookup Engine
20
L0 Storage
21
L1 Storage
22
L2 Storage
23
L3 Storage
24
30
L4 Associated Data
25
30
L5 Associated Data
26
Result Buffer
62
43
44

FIG. 2

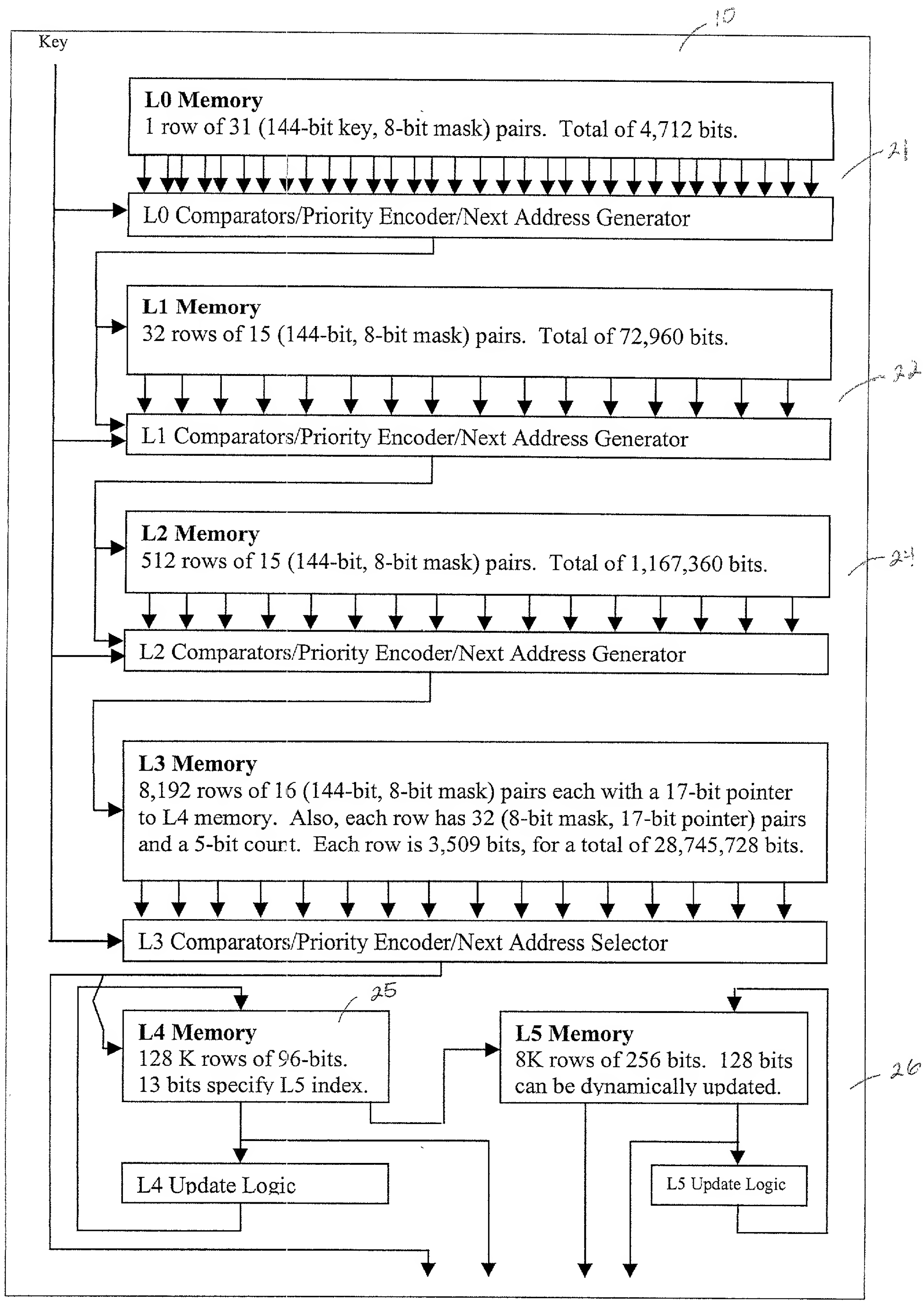
10
Key
L0 Memory
1 row of 31 (144-bit key, 8-bit mask) pairs. Total of 4,712 bits.
21
L0 Comparators/Priority Encoder/Next Address Generator
L1 Memory
32 rows of 15 (144-bit, 8-bit mask) pairs. Total of 72,960 bits.
22
L1 Comparators/Priority Encoder/Next Address Generator
L2 Memory
512 rows of 15 (144-bit, 8-bit mask) pairs. Total of 1,167,360 bits.
24
L2 Comparators/Priority Encoder/Next Address Generator
L3 Memory
8,192 rows of 16 (144-bit, 8-bit mask) pairs each with a 17-bit pointer to L4 memory. Also, each row has 32 (8-bit mask, 17-bit pointer) pairs and a 5-bit count. Each row is 3,509 bits, for a total of 28,745,728 bits.
L3 Comparators/Priority Encoder/Next Address Selector
25
L4 Memory
128 K rows of 96-bits.
13 bits specify L5 index.
L5 Memory
8K rows of 256 bits. 128 bits can be dynamically updated.
26
L4 Update Logic
L5 Update Logic

FIG. 3

DECLARATION AND POWER OF ATTORNEY

As a below named inventor, I hereby declare that:

My residence, postal address and citizenship are as stated below next to my name.

I believe that I am the original, first and sole inventor (if only one name is listed below) or an original first and joint inventor (if multiple names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

A HIGH PERFORMANCE NETWORK ADDRESS PROCESSOR SYSTEM

the specification of which is attached hereto unless the following information is indicated:
_____ was filed on ______________;
as United States Application Number or PCT International Application Number ___________;
and was amended on ______________________ (if applicable).

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendments referred to above.

I acknowledge the duty to disclose information that is material to patentability as defined in 37 CFR 1.56.

I hereby claim foreign priority benefits under 35 U.S.C. 119(a)-(d) or 365(b) of any foreign application(s) for patent or inventor's certificate, or 365(a) of any PCT International application which designated at least one country other than the United States, listed below and have also identified, as so indicated below, any foreign application for patent or inventor's certificate, or PCT International application having a filing date before that of the application on which priority is claimed.

Prior Foreign Application(s) Priority Claimed

______________	______________	______________	___ Yes ___ No
(Application No.)	(Country)	(Day/Month/Year Filed)	

I hereby claim the benefit under 35 U.S.C. 119(e) of any United States provisional application(s) listed below.

______________	______________
(Application No.)	(Day/Month/Year Filed)

I hereby claim the benefit under 35 U.S.C. 120 of any United States application(s), or 365(c) of any PCT International application designating the United States, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or PCT International application in the manner provided by the first paragraph of 35 U.S.C. 112, I acknowledge the duty to disclose information that is material to patentability as defined in 37 C.F.R. 1.56 that became available between the filing date of the prior application and the national or PCT International filing date of this application.

______________	______________	______________
(Application No.)	(Day/Month/Year Filed)	(Status - patented, pending, abandoned)

I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and to transact all business in the United States Patent and Trademark Office connected therewith:

DENNIS S. FERNANDEZ, REG. NO. *34,160*
IRENE H. FERNANDEZ, REG. NO. *34,625*
PETER C. SU, REG. NO. *43,939*

Send Correspondence and Communications to:

FERNANDEZ & ASSOCIATES, LLP
PATENT ATTORNEYS
PO BOX D
MENLO PARK, CA 94026-6204
CUSTOMER NUMBER: 22877
(650) 325-4999
(650) 325-1203 : FAX
EMAIL: *iploft@iploft.com*

022877
PATENT TRADEMARK OFFICE

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and beliefs are believed to be true; and further that these statements were made with knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full name of sole or first inventor: **Jonathan Lockwood Huie**	
Inventor's signature Jonathan Lockwood Huie	Date 6/9/2000
Residence: **65 E. Latimer Ave, Campbell, CA 95008**	
Citizenship: **United States**	
Postal Address: **SAME AS RESIDENCE**	

Full name of joint inventor: **James Michael O'Connor**	
Inventor's signature J M. O'Connor	Date 6/8/2000
Residence: **1877 Slate Drive, Union City, CA 94587**	
Citizenship: **United States**	
Postal Address: **SAME AS RESIDENCE**	